Problem 1 – Using Make

In previous labs, you invoked the compiler directly using the *gcc* command. As your code gets more complicated, this can be kind of a pain. From now on, you will be using the *make* command to compile your code.

Re-write your "hello world" program (or just copy and paste it) from week 1, but instead of running *gcc*, use *make* to compile it. For example, if you have source code in your directory called *hello.c* run

```
make hello
```

from the command line. When you run that command, you're telling make that you want a file named hello to be created, then make does the following:

1. Does the file *hello* exist already?
2. No. Ok, is there another file which starts with *hello*?
3. Yes, it's called *hello.c*, Do I know how to build *.c* files?
4. Yes, I run the command *cc hello.c –o hello* to build them
5. I will run that command to build the executable *hello*

When you run *make*, it will create an executable with the same name as the source code. Did you name your file *lab1a.c*? The executable will be named *lab1a*. There is much more you can do with *make*, but for now this will suffice.

Problem 2

Assume the user has entered a number with *n* digits. Assume the maximum number that can be entered is 2,147,483,647. The first line is to start with the leftmost digit and print *n* digits; the second line is to start with the second digit from the left and print *n-1* digits, and so forth. This is repeated again, but this time inversely to create an hour-glass shape. **This program must utilize loops and logic to extract the exact amount of digits entered.**

```
1  2  3  4
 2  3  4
   3  4
     4
   4  3
 4  3  2
4  3  2  1
```